

Doxxygen_MPDASTARONLINE_DB Reference Manual

Generated by Doxygen 1.3.7

Wed Sep 6 11:51:06 2006

Contents

1	Doxygen_MPDAUTHENTICATION Class Index	1
1.1	Doxygen_MPDAUTHENTICATION Class List	1
2	Doxygen_MPDAUTHENTICATION File Index	3
2.1	Doxygen_MPDAUTHENTICATION File List	3
3	Doxygen_MPDAUTHENTICATION Class Documentation	5
3.1	bbchvSender Class Reference	5
3.2	fpdhvSender Class Reference	16
4	Doxygen_MPDAUTHENTICATION File Documentation	27
4.1	bbchvDaemon.cc File Reference	27
4.2	bbchvSender.cc File Reference	28
4.3	bbchvSender.hh File Reference	29
4.4	bbchvSender_i.cc File Reference	30
4.5	fpdhvDaemon.cc File Reference	31
4.6	fpdhvSender.cc File Reference	32
4.7	fpdhvSender.hh File Reference	33
4.8	fpdhvSender_i.cc File Reference	34

Chapter 1

Doxxygen_MP_D_STAR_ONLINE_DB Class Index

1.1 Doxygen_MP_D_STAR_ONLINE_DB Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

bbchvSender	5
fpdhvSender	16

Chapter 2

Doxxygen_MPDAEMON_StarOnlineDB File Index

2.1 Doxygen_MPDAEMON_StarOnlineDB File List

Here is a list of all files with brief descriptions:

bbchvDaemon.cc	27
bbchvSender.cc	28
bbchvSender.hh	29
bbchvSender_i.cc	30
fpdhvDaemon.cc	31
fpdhvSender.cc	32
fpdhvSender.hh	33
fpdhvSender_i.cc	34

Chapter 3

Doxxygen_MP_D_STAR_ONLINE_DB Class Documentation

3.1 bbchvSender Class Reference

```
#include <bbchvSender.hh>
```

Public Member Functions

- `bbchvSender` (const char *localDir)
- virtual `~bbchvSender` ()
- virtual void `initTable` ()
- virtual void `initTags` ()
- virtual void `initDataBase` ()
- virtual bool `loadUserControls` (const char *name, const char *value)
- virtual void `initQuery` ()
- virtual bool `queryData` ()
- virtual bool `readData` (const char *fileName)
- virtual bool `updateDb` (const char *fileName)
- virtual bool `readData` (ifstream &from)
- virtual bool `hasChanged` (int rowNumber)
- char * `readAny` ()
- bool `readVal` (char *&value)
- bool `readVal` (float &value)
- bool `readVal` (double &value)
- bool `readVal` (short &value)
- bool `readVal` (int &value)
- bool `readVal` (long &value)
- bool `readVal` (long long &value)
- bool `nextLine` (ifstream &from)
- void `readError` (int l, char *c, char *m)

Protected Attributes

- bbchv `previousVals` [NUM_DB_ROWS]
- bbchv `tempVals` [NUM_DB_ROWS]
- int `elementList` [NUM_DB_ROWS]
- bbchv `updateVals` [NUM_DB_ROWS]
- int `updateElements` [NUM_DB_ROWS]
- bool `mreadStatus`
- char `mline` [256]
- char `tpline` [256]
- char * `ptr1`
- char * `ptr2`
- float `vdriftLimit`
dito
- float `cdriftLimit`

3.1.1 Constructor & Destructor Documentation

3.1.1.1 `bbchvSender::bbchvSender (const char * localDir)`

Definition at line 19 of file bbchvSender.cc.

```
19
20
21     initTags();
22     if(localDir) cd(localDir); // note this ignores the sub dir tag
23     init("bbchv"); // setup the file I/O
24     initDataBase(); // database connections
25     initTable(); // table definitions
26
27 }
```

3.1.1.2 `virtual bbchvSender::~bbchvSender () [inline, virtual]`

Definition at line 42 of file bbchvSender.hh.

```
42 {};
```

3.1.2 Member Function Documentation

3.1.2.1 `bool bbchvSender::hasChanged (int rowNumber) [virtual]`

Definition at line 113 of file bbchvSender_i.cc.

```
113
114
115 bbchv* pre=&previousVals[rowNumber];
116 bbchv* cur=&tempVals[rowNumber];
117
118 if((fabs(pre->demand-cur->demand)>=vdriftLimit))return true;
119 if((fabs(pre->measured-cur->measured)>=vdriftLimit))return true;
```

```

120 if((fabs(pre->current-cur->current)>=cdriftLimit))return true;
121
122 /* example ... note -> change to any element requires db-update
123 * and thus returns true immediately
124 *
125 *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
126 *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
127 *
128 * ...
129 */
130
131 return false;
132 }
```

3.1.2.2 void bbchvSender::initDataBase () [virtual]

Definition at line 75 of file bbchvSender.cc.

```

75 {
76 #define __METHOD__ "initDataBase()"
77
78 /* More than an example... swap user & dbTrg as per subsystem*/
79 mgr->setUser("stardb","");
80 StDbType dbT = dbConditions;
81 StDbDomain dbD = dbBbc;
82
83 if( !( node = mgr->initConfig(dbT,dbD)) )
84     sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
85
86 #undef __METHOD__
87 }
```

3.1.2.3 void bbchvSender::initQuery () [virtual]

Definition at line 41 of file bbchvSender_i.cc.

```

41 {
42 #define __METHOD__ "initQuery()"
43
44 ofstream to(queryFile);
45
46 if(!to.is_open()){
47     sendMess("Open Failed ",queryFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
48     return;
49 }
50
51 char* bases[]={ "BBChv:SUB_SET_DMD", "BBChv:SUB_RD_V", "BBChv:SUB_RD_I" };
52 char* chlabel[]={ "E", "F", "G", "H", "I", "J", "K", "L" };
53 for(int is=0;is<4;is++){
54     for(int ic=0;ic<8;ic++){
55         for(int j=0;j<3;j++) to<<bases[j]<<"_1:"<<is<<". "<<chlabel[ic]<<endl;
56         for(int ic=0;ic<8;ic++)
57             for(int j=0;j<3;j++) to<<bases[j]<<"1_1:"<<is<<". "<<chlabel[ic]<<endl;
58     };
59
60 /* example
61 *      for(int i=0;i<16;i++){
62 *          to<<"TRGhv:SUB_RD_V_1:<<i<<".E"<<endl;
63 *          to<<"TRGhv:SUB_RD_V_1:<<i<<".F"<<endl;
64 *          ...
65 *
```

```

66   *
67   */
68
69     to.close();
70
71 #undef __METHOD__
72 }
```

3.1.2.4 void bbchvSender::initTable () [virtual]

Definition at line 30 of file bbchvSender.cc.

```

30           {
31 #define __METHOD__ "initTable()"
32
33   StDbTable* table=0;
34   if(!(table=node->addDbTable("bbchv")))
35     sendMess("Could not find table=bbchv",dbMFatal,__LINE__,__CLASS__,__METHOD__);
36
37 memset(tempVals,0,NUM_DB_ROWS*sizeof(bbchv));
38 memset(previousVals,0,NUM_DB_ROWS*sizeof(bbchv));
39
40   int nrows;
41   int* elist = table->getElementID(nrows);
42   if(nrows!=NUM_DB_ROWS){
43     //char mess[256];
44     ostringstream ms;
45     ms<<"Db rows("<<nrows<<") != compiled("<<NUM_DB_ROWS<<") "<<ends;
46     sendMess((ms.str()).c_str(),dbMFatal,__LINE__,__CLASS__,__METHOD__);
47   }
48   memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
49
50   unsigned int timestamp=time(NULL);
51   mgr->setRequestTime(timestamp);
52   if(mgr->fetchDbTable(table)){
53     bbchv* thv = (bbchv*)table->GetTable();
54     memcpy(previousVals,thv,nrows*sizeof(bbchv));
55   }
56
57 #undef __METHOD__
58 };
```

3.1.2.5 void bbchvSender::initTags () [virtual]

Definition at line 66 of file bbchvSender.cc.

```

66           {
67   /* more than an example -> swap "trg" to your subsys & add to email list*/
68   setEmailTo("porter@bnl.gov");
69   setDomainName("bbc");
70
71 }
```

3.1.2.6 bool bbchvSender::loadUserControls (const char * name, const char * value) [virtual]

Definition at line 19 of file bbchvSender_i.cc.

```

19
20 #define __METHOD__ "loadUserControls(name,value)"
21
22 /* more than an example ... swap driftLimit to yours
23 * and duplicate this structure for each selection criteria */
24 if(strstr(name,"vdriftLimit")){
25     vdriftLimit=atof(value);
26     sendMess("vdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
27     return true;
28 }
29 if(strstr(name,"cdriftLimit")){
30     cdriftLimit=atof(value);
31     sendMess("cdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
32     return true;
33 }
34
35 return false;
36 #undef __METHOD__
37 }
```

3.1.2.7 bool bbchvSender::nextLine (ifstream &from) [inline]

Definition at line 75 of file bbchvSender.hh.

```

75
76     if(!from.getline(mline,255))return false;
77     return true;
78 }
```

3.1.2.8 bool bbchvSender::queryData () [virtual]

Definition at line 91 of file bbchvSender.cc.

```

91
92 #define __METHOD__ "queryData()"
93
94 /*
95  * MORE THAN AN EXAMPLE....
96  * IF Standard SC-Query via "caGet" then,
97  * no need to change this method AT ALL
98  *
99 */
100
101 writeTime = (unsigned int)time(NULL);           //for database write time
102
103 //char systemCmd[1024];
104 ostringstream scmd;
105 scmd<<"caGet "<<queryFile<<" "<<dataFile<<ends;
106
107 if(system((scmd.str()).c_str()))
108     return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
109
110 return true;
111 #undef __METHOD__
112 };
```

3.1.2.9 char * bbchvSender::readAny ()

Definition at line 197 of file bbchvSender.cc.

```

197
198
199     strcpy(tmpLine,mLine);
200     ptr1=tmpLine;
201     ptr2=strtok(ptr1," ");
202     if(!ptr2) return ptr2;
203     ptr2=strtok(NULL," ");
204     return ptr2;
205 }
```

3.1.2.10 bool bbchvSender::readData (ifstream &from) [virtual]

Definition at line 77 of file bbchvSender_i.cc.

```

77
78 #define __METHOD__ "readData(ifstream)"
79
80 mreadStatus=true;
81 memset(tempVals,0,NUM_DB_ROWS*sizeof(bbchv));
82
83
84 int i=0;
85 char* c=__CLASS__;
86 char* m=__METHOD__;
87
88 for(int is=0;is<4;is++){
89     for(int ic=0;ic<16;ic++){
90         tempVals[i].slot=is;
91         tempVals[i].channel=ic;
92         if(!nextLine(from) || !readVal(tempVals[i].demand))readError(__LINE__,c,m);      if(!nextLine(from)
93         if(!nextLine(from) || !readVal(tempVals[i].current))readError(__LINE__,c,m);
94     i++;
95 }
96 }
97 /* example format
98 *   for(int i=0;i<NUM_DB_ROWS;i++){
99 *     if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(l,c,m);
100 *
101 *     ....
102 *
103 *   }
104 */
105
106     from.close();
107 return true;
108 #undef __METHOD__
109 }
```

3.1.2.11 bool bbchvSender::readData (const char *fileName) [virtual]

Definition at line 116 of file bbchvSender.cc.

```

116
117 #define __METHOD__ "readData(fileName)"
118
119     ifstream from(fileName);
120     if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
121
122     return readData(from); // user implemented file read
123 #undef __METHOD__
124 }
```

3.1.2.12 void bbchvSender::readError (int *l*, char * *c*, char * *m*) [inline]

Definition at line 80 of file bbchvSender.hh.

```
80
81     mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
82 }
```

3.1.2.13 bool bbchvSender::readVal (long long & *value*)

Definition at line 269 of file bbchvSender.cc.

```
269
270
271     if(!readAny())return false;
272     char* store[256];
273     value=strtoll(ptr2,store,10);
274     if(strlen(*store)>0) return false; // value is not a number
275
276     return true;
277 };
```

3.1.2.14 bool bbchvSender::readVal (long & *value*)

Definition at line 258 of file bbchvSender.cc.

```
258
259
260     if(!readAny())return false;
261
262     char* store[256];
263     value=strtol(ptr2,store,10);
264     if(strlen(*store)>0) return false; // value is not a number
265
266     return true;
267 };
```

3.1.2.15 bool bbchvSender::readVal (int & *value*)

Definition at line 247 of file bbchvSender.cc.

```
247
248
249     if(!readAny()) return false;
250
251     char* store[256];
252     value=(int)strtol(ptr2,store,10);
253     if(strlen(*store)>0) return false; // value is not a number
254
255     return true;
256 };
```

3.1.2.16 bool bbchvSender::readVal (short & *value*)

Definition at line 236 of file bbchvSender.cc.

```
236
237
238     if(!readAny()) return false;
239
240     char* store[256];
241     value=(short)strtol(ptr2,store,10);
242     if(strlen(*store)>0) return false; // value is not a number
243
244     return true;
245 };
```

3.1.2.17 bool bbchvSender::readVal (double & *value*)

Definition at line 225 of file bbchvSender.cc.

```
225
226
227     if(!readAny())return false;
228
229     char* store[256];
230     value=strtod(ptr2,store);
231     if(strlen(*store)>0) return false; // value is not a number
232
233     return true;
234 };
```

3.1.2.18 bool bbchvSender::readVal (float & *value*)

Definition at line 214 of file bbchvSender.cc.

```
214
215
216     if(!readAny()) return false;
217
218     char* store[256];
219     value=(float)strtod(ptr2,store);
220     if(strlen(*store)>0) return false; // value is not a number
221
222     return true;
223 };
```

3.1.2.19 bool bbchvSender::readVal (char *& *value*)

Definition at line 207 of file bbchvSender.cc.

```
207
208
209     if(!readAny()) return false;
210     strcpy(value,ptr2);
211     return true;
212 }
```

3.1.2.20 bool bbchvSender::updateDb (const char **fileName*) [virtual]

Definition at line 127 of file bbchvSender.cc.

```

127
128 #define __METHOD__ "updateDb(filename)"
129
130     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
131
132     int* elements;
133     bbchv* vals;
134     int numRows = 0;
135
136     if(writeRequired()){
137
138         numRows=NUM_DB_ROWS;
139         elements=elementList;
140         vals = tempVals;
141
142     } else {
143
144         for(int i=0; i<NUM_DB_ROWS; i++){
145             if(hasChanged(i)){
146                 updateElements[numRows]=elementList[i];
147                 updateVals[numRows] = tempVals[i];
148                 previousVals[i]=tempVals[i];
149                 numRows++;
150             }
151         }
152
153         elements = updateElements;
154         vals      = updateVals;
155     }
156
157     if(numRows==0) return sendMess(" No update required for",mbaseName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
158
159 //char mess[256];
160 ostringstream sn;
161 sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows "<<ends;
162 sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
163
164 StDbTable* dbTable=node->findTable("bbchv");
165 dbTable->SetTable((char*)vals, numRows, elements);
166 mgr->setStoreTime(writeTime);
167
168     if(!mgr->storeDbTable(dbTable)) {
169         addBackLog(writeTime);
170         return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
171     }
172
173     if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
174
175     return true;
176 #undef __METHOD__
177 }
```

3.1.3 Member Data Documentation**3.1.3.1 float bbchvSender::cdriftLimit [protected]**

Definition at line 36 of file bbchvSender.hh.

3.1.3.2 int bbchvSender::elementList[NUM_DB_ROWS] [protected]

Definition at line 24 of file bbchvSender.hh.

3.1.3.3 char bbchvSender::mline[256] [protected]

Definition at line 29 of file bbchvSender.hh.

3.1.3.4 bool bbchvSender::mreadStatus [protected]

Definition at line 28 of file bbchvSender.hh.

3.1.3.5 bbchv bbchvSender::previousVals[NUM_DB_ROWS] [protected]

Definition at line 22 of file bbchvSender.hh.

3.1.3.6 char* bbchvSender::ptr1 [protected]

Definition at line 31 of file bbchvSender.hh.

3.1.3.7 char * bbchvSender::ptr2 [protected]

Definition at line 31 of file bbchvSender.hh.

3.1.3.8 bbchv bbchvSender::tempVals[NUM_DB_ROWS] [protected]

Definition at line 23 of file bbchvSender.hh.

3.1.3.9 char bbchvSender::tmpline[256] [protected]

Definition at line 30 of file bbchvSender.hh.

3.1.3.10 int bbchvSender::updateElements[NUM_DB_ROWS] [protected]

Definition at line 26 of file bbchvSender.hh.

3.1.3.11 bbchv bbchvSender::updateVals[NUM_DB_ROWS] [protected]

Definition at line 25 of file bbchvSender.hh.

3.1.3.12 float bbchvSender::vdriftLimit [protected]

dito

Definition at line 35 of file bbchvSender.hh.

The documentation for this class was generated from the following files:

- [bbchvSender.hh](#)
- [bbchvSender.cc](#)
- [bbchvSender_i.cc](#)

3.2 fpdhvSender Class Reference

```
#include <fpdhvSender.hh>
```

Public Member Functions

- `fpdhvSender (const char *localDir)`
- `virtual ~fpdhvSender ()`
- `virtual void initTable ()`
- `virtual void initTags ()`
- `virtual void initDataBase ()`
- `virtual bool loadUserControls (const char *name, const char *value)`
- `virtual void initQuery ()`
- `virtual bool queryData ()`
- `virtual bool readData (const char *fileName)`
- `virtual bool updateDb (const char *fileName)`
- `virtual bool readData (ifstream &from)`
- `virtual bool hasChanged (int rowNumber)`
- `char * readAny ()`
- `bool readVal (char *&value)`
- `bool readVal (float &value)`
- `bool readVal (double &value)`
- `bool readVal (short &value)`
- `bool readVal (int &value)`
- `bool readVal (long &value)`
- `bool readVal (long long &value)`
- `bool nextLine (ifstream &from)`
- `void readError (int l, char *c, char *m)`

Protected Attributes

- `fpdhv previousVals [NUM_DB_ROWS]`
- `fpdhv tempVals [NUM_DB_ROWS]`
- `int elementList [NUM_DB_ROWS]`
- `fpdhv updateVals [NUM_DB_ROWS]`
- `int updateElements [NUM_DB_ROWS]`
- `bool mreadStatus`
- `char mline [256]`
- `char tmpline [256]`
- `char * ptr1`
- `char * ptr2`
- `float vdriftLimit`
dito
- `float cdriftLimit`

3.2.1 Constructor & Destructor Documentation

3.2.1.1 fpdhvSender::fpdhvSender (const char * *localDir*)

Definition at line 19 of file fpdhvSender.cc.

```

19
20
21     initTags();
22     if(localDir) cd(localDir); // note this ignores the sub dir tag
23     init("fpdhv"); // setup the file I/O
24     initDataBase(); // database connections
25     initTable(); // table definitions
26
27 }
```

3.2.1.2 virtual fpdhvSender::~fpdhvSender () [inline, virtual]

Definition at line 42 of file fpdhvSender.hh.

```
42 {};
```

3.2.2 Member Function Documentation

3.2.2.1 bool fpdhvSender::hasChanged (int *rowNumber*) [virtual]

Definition at line 129 of file fpdhvSender_i.cc.

```

129
130
131 fpdhv* pre=&previousVals[rowNumber];
132 fpdhv* cur=&tempVals[rowNumber];
133
134 if((fabs(pre->demand-cur->demand)>=vdriftLimit))return true;
135 if((fabs(pre->measured-cur->measured)>=vdriftLimit))return true;
136 if((fabs(pre->current-cur->current)>=cdriftLimit))return true;
137
138 /* example ... note -> change to any element requires db-update
139 * and thus returns true immediately
140 */
141 *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
142 *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
143 *
144 * ....
145 */
146
147 return false;
148 }
```

3.2.2.2 void fpdhvSender::initDataBase () [virtual]

Definition at line 75 of file fpdhvSender.cc.

```

75
76 #define __METHOD__ "initDataBase()"
```

```

77
78     /* More than an example... swap user & dbTrg as per subsystem*/
79     mgr->setUser("stardb","");
80     StDbType    dbT = dbConditions;
81     StDbDomain dbD = dbBbc;
82
83     if( !( node = mgr->initConfig(dbT,dbD) ) )
84         sendMess( "Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
85
86 #undef __METHOD__
87 }
```

3.2.2.3 void fpdhvSender::initQuery () [virtual]

Definition at line 42 of file fpdhvSender_i.cc.

```

42
43 #define __METHOD__ "initQuery()"
44
45     ofstream to(queryFile);
46
47     if(!to.is_open()){
48         sendMess( "Open Failed ",queryFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
49         return;
50     }
51
52     char* Ebases[]={"FDEhv:SUB_SET_DMD","FDEhv:SUB_RD_V","FDEhv:SUB_RD_I"};
53     char* Wbases[]={"FDWhv:SUB_SET_DMD","FDWhv:SUB_RD_V","FDWhv:SUB_RD_I"};
54     char* chlabel[]={"E","F","G","H","I","J","K","L"};
55     for(int is=0;is<12;is++){
56         for(int ic=0;ic<8;ic++)
57             for(int j=0;j<3;j++) to<<Ebases[j]<<"_1:<<is<<."<<chlabel[ic]<<endl;
58         for(int ic=0;ic<8;ic++)
59             for(int j=0;j<3;j++) to<<Ebases[j]<<"1_1:<<is<<."<<chlabel[ic]<<endl;
60     }
61
62     for(int is=0;is<16;is++){
63         if((is!=15) && (is>5)) continue;
64         for(int ic=0;ic<8;ic++)
65             for(int j=0;j<3;j++) to<<Wbases[j]<<"_1:<<is<<."<<chlabel[ic]<<endl;
66         for(int ic=0;ic<8;ic++)
67             for(int j=0;j<3;j++) to<<Wbases[j]<<"1_1:<<is<<."<<chlabel[ic]<<endl;
68     }
69
70
71
72 /* example
73 *      for(int i=0;i<16;i++){
74 *          to<<"TRGhv:SUB_RD_V_1:<<i<<".E"<<endl;
75 *          to<<"TRGhv:SUB_RD_V_1:<<i<<".F"<<endl;
76 *          ....
77 *
78 *      }
79 */
80
81     to.close();
82
83 #undef __METHOD__
84 }
```

3.2.2.4 void fpdhvSender::initTable () [virtual]

Definition at line 30 of file fpdhvSender.cc.

```

30             {
31 #define __METHOD__ "initTable()"
32
33     StDbTable* table=0;
34     if(!(table=node->addDbTable("fpdhv")))
35         sendMess("Could not find table=fpdhv",dbMFatal,__LINE__,__CLASS__,__METHOD__);
36
37     memset(tempVals,0,NUM_DB_ROWS*sizeof(fpdhv));
38     memset(previousVals,0,NUM_DB_ROWS*sizeof(fpdhv));
39
40     int nrows;
41     int* elist = table->getElementID(nrows);
42     if(nrows!=NUM_DB_ROWS){
43         //char mess[256];
44         ostringstream ms;
45         ms<<"Db rows("<<nrows<<") != compiled(";<<NUM_DB_ROWS;<<") "<<ends;
46         sendMess(ms.str().c_str(),dbMFatal,__LINE__,__CLASS__,__METHOD__);
47     }
48     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
49
50     unsigned int timestamp=time(NULL);
51     mgr->setRequestTime(timestamp);
52     if(mgr->fetchDbTable(table)){
53         fpdhv* thv = (fpdhv*)table->GetTable();
54         memcpy(previousVals,thv,nrows*sizeof(fpdhv));
55     }
56
57 #undef __METHOD__
58 };

```

3.2.2.5 void fpdhvSender::initTags () [virtual]

Definition at line 66 of file fpdhvSender.cc.

```

66             {
67     /* more than an example -> swap "trg" to your subsys & add to email list*/
68     setEmailTo("porter@bnl.gov");
69     setDomainName("bbc");
70
71 }

```

3.2.2.6 bool fpdhvSender::loadUserControls (const char * name, const char * value) [virtual]

Definition at line 20 of file fpdhvSender_i.cc.

```

20             {
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24 * and duplicate this structure for each selection criteria */
25     if strstr(name,"vdriftLimit")){
26         vdriftLimit=atof(value);
27         sendMess("vdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
28         return true;
29     }
30     if strstr(name,"cdriftLimit")){
31         cdriftLimit=atof(value);
32         sendMess("cdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
33         return true;
34     }
35

```

```
36 return false;
37 #undef __METHOD__
38 }
```

3.2.2.7 bool fpdhvSender::nextLine (ifstream & from) [inline]

Definition at line 75 of file fpdhvSender.hh.

```
75 {
76     if(!from.getline(mline,255))return false;
77     return true;
78 }
```

3.2.2.8 bool fpdhvSender::queryData () [virtual]

Definition at line 91 of file fpdhvSender.cc.

```
91             {
92 #define __METHOD__ "queryData()"
93
94     /*
95      * MORE THAN AN EXAMPLE....
96      * IF Standard SC-Query via "caGet" then,
97      * no need to change this method AT ALL
98      *
99     */
100
101    writeTime = (unsigned int)time(NULL);           //for database write time
102
103 //   char systemCmd[1024];
104   ostringstream scmd;
105   scmd<<"caGet "<<queryFile<<" "<<dataFile<<ends;
106
107   if(system((scmd.str()).c_str()))
108       return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
109
110   return true;
111 #undef __METHOD__
112 };
```

3.2.2.9 char * fpdhvSender::readAny ()

Definition at line 197 of file fpdhvSender.cc.

```
197             {
198
199     strcpy(tmpLine,mline);
200     ptr1=tmpLine;
201     ptr2=strtok(ptr1," ");
202     if(!ptr2) return ptr2;
203     ptr2=strtok(NULL," ");
204     return ptr2;
205 }
```

3.2.2.10 bool fpdhvSender::readData (ifstream &from) [virtual]

Definition at line 89 of file fpdhvSender_i.cc.

```

89
90 #define __METHOD__ "readData(ifstream)"
91
92 mreadStatus=true;
93 memset(tempVals,0,NUM_DB_ROWS*sizeof(fpdhv));
94
95
96 int i=0;
97 char* c=__CLASS__;
98 char* m=__METHOD__;
99
100
101 for(int is=0;is<12;is++){
102   for(int ic=0;ic<16;ic++){
103     tempVals[i].slot=is;
104     tempVals[i].channel=ic;
105     if(!nextLine(from) || !readVal(tempVals[i].demand))readError(__LINE__,c,m);      if(!nextLine(from)
106     if(!nextLine(from) || !readVal(tempVals[i].current))readError(__LINE__,c,m);
107     i++;
108   }
109 }
110 for(int is=0;is<16;is++){
111   if((is!=15) && (is>5))continue;
112   for(int ic=0;ic<16;ic++){
113     tempVals[i].slot=is;
114     tempVals[i].channel=ic;
115     if(!nextLine(from) || !readVal(tempVals[i].demand))readError(__LINE__,c,m);      if(!nextLine(from)
116     if(!nextLine(from) || !readVal(tempVals[i].current))readError(__LINE__,c,m);
117     i++;
118   }
119 }
120
121
122   from.close();
123 return true;
124 #undef __METHOD__
125 }
```

3.2.2.11 bool fpdhvSender::readData (const char *fileName) [virtual]

Definition at line 116 of file fpdhvSender.cc.

```

116
117 #define __METHOD__ "readData(fileName)"
118
119   ifstream from(fileName);
120   if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
121
122   return readData(from); // user implemented file read
123 #undef __METHOD__
124 }
```

3.2.2.12 void fpdhvSender::readError (int l, char *c, char *m) [inline]

Definition at line 80 of file fpdhvSender.hh.

```

80
81   mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
82 }
```

3.2.2.13 bool fpdhvSender::readVal (long long & value)

Definition at line 269 of file fpdhvSender.cc.

```

269
270
271   if(!readAny())return false;
272   char* store[256];
273   value=strtoll(ptr2,store,10);
274   if(strlen(*store)>0) return false; // value is not a number
275
276   return true;
277 };
```

3.2.2.14 bool fpdhvSender::readVal (long & value)

Definition at line 258 of file fpdhvSender.cc.

```

258
259
260   if(!readAny())return false;
261
262   char* store[256];
263   value=strtol(ptr2,store,10);
264   if(strlen(*store)>0) return false; // value is not a number
265
266   return true;
267 };
```

3.2.2.15 bool fpdhvSender::readVal (int & value)

Definition at line 247 of file fpdhvSender.cc.

```

247
248
249   if(!readAny()) return false;
250
251   char* store[256];
252   value=(int)strtol(ptr2,store,10);
253   if(strlen(*store)>0) return false; // value is not a number
254
255   return true;
256 };
```

3.2.2.16 bool fpdhvSender::readVal (short & value)

Definition at line 236 of file fpdhvSender.cc.

```

236
237
238     if(!readAny()) return false;
239
240     char* store[256];
241     value=(short)strtol(ptr2,store,10);
242     if(strlen(*store)>0) return false; // value is not a number
243
244     return true;
245 };

```

3.2.2.17 bool fpdhvSender::readVal (double & *value*)

Definition at line 225 of file fpdhvSender.cc.

```

225
226
227     if(!readAny())return false;
228
229     char* store[256];
230     value=strtod(ptr2,store);
231     if(strlen(*store)>0) return false; // value is not a number
232
233     return true;
234 };

```

3.2.2.18 bool fpdhvSender::readVal (float & *value*)

Definition at line 214 of file fpdhvSender.cc.

```

214
215
216     if(!readAny()) return false;
217
218     char* store[256];
219     value=(float)strtod(ptr2,store);
220     if(strlen(*store)>0) return false; // value is not a number
221
222     return true;
223 };

```

3.2.2.19 bool fpdhvSender::readVal (char *& *value*)

Definition at line 207 of file fpdhvSender.cc.

```

207
208
209     if(!readAny()) return false;
210     strcpy(value,ptr2);
211     return true;
212 }

```

3.2.2.20 bool fpdhvSender::updateDb (const char **fileName*) [virtual]

Definition at line 127 of file fpdhvSender.cc.

```

127
128 #define __METHOD__ "updateDb(filename)"
129
130     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
131
132     int* elements;
133     fpdhv* vals;
134     int numRows = 0;
135
136     if(writeRequired()){
137
138         numRows=NUM_DB_ROWS;
139         elements=elementList;
140         vals = tempVals;
141
142     } else {
143
144         for(int i=0; i<NUM_DB_ROWS; i++){
145             if(hasChanged(i)){
146                 updateElements[numRows]=elementList[i];
147                 updateVals[numRows] = tempVals[i];
148                 previousVals[i]=tempVals[i];
149                 numRows++;
150             }
151         }
152
153         elements = updateElements;
154         vals      = updateVals;
155     }
156
157     if(numRows==0) return sendMess(" No update required for",mbaseName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
158
159 //char mess[256];
160 ostringstream sn;
161 sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows "<<ends;
162 sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
163
164 StDbTable* dbTable=node->findTable("fpdhv");
165 dbTable->SetTable((char*)vals, numRows, elements);
166 mgr->setStoreTime(writeTime);
167
168     if(!mgr->storeDbTable(dbTable)) {
169         addBackLog(writeTime);
170         return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
171     }
172
173     if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
174
175     return true;
176 #undef __METHOD__
177 }
```

3.2.3 Member Data Documentation

3.2.3.1 float fpdhvSender::cdriftLimit [protected]

Definition at line 36 of file fpdhvSender.hh.

3.2.3.2 int fpdhvSender::elementList[NUM_DB_ROWS] [protected]

Definition at line 24 of file fpdhvSender.hh.

3.2.3.3 char fpdhvSender::mline[256] [protected]

Definition at line 29 of file fpdhvSender.hh.

3.2.3.4 bool fpdhvSender::mreadStatus [protected]

Definition at line 28 of file fpdhvSender.hh.

3.2.3.5 fpdhv fpdhvSender::previousVals[NUM_DB_ROWS] [protected]

Definition at line 22 of file fpdhvSender.hh.

3.2.3.6 char* fpdhvSender::ptr1 [protected]

Definition at line 31 of file fpdhvSender.hh.

3.2.3.7 char * fpdhvSender::ptr2 [protected]

Definition at line 31 of file fpdhvSender.hh.

3.2.3.8 fpdhv fpdhvSender::tempVals[NUM_DB_ROWS] [protected]

Definition at line 23 of file fpdhvSender.hh.

3.2.3.9 char fpdhvSender::ttmpLine[256] [protected]

Definition at line 30 of file fpdhvSender.hh.

3.2.3.10 int fpdhvSender::updateElements[NUM_DB_ROWS] [protected]

Definition at line 26 of file fpdhvSender.hh.

3.2.3.11 fpdhv fpdhvSender::updateVals[NUM_DB_ROWS] [protected]

Definition at line 25 of file fpdhvSender.hh.

3.2.3.12 float fpdhvSender::vdriftLimit [protected]

dito

Definition at line 35 of file fpdhvSender.hh.

The documentation for this class was generated from the following files:

- [fpdhvSender.hh](#)
- [fpdhvSender.cc](#)
- [fpdhvSender_i.cc](#)

Chapter 4

Doxxygen_MPDAEMON_STAR_ONLINE_DB File Documentation

4.1 bbchvDaemon.cc File Reference

```
#include "bbchvSender.hh"
#include <unistd.h>
```

Functions

- void [runSender](#) (const char **ldir*)

4.1.1 Function Documentation

4.1.1.1 void [runSender](#) (const char * *ldir*)

Definition at line 14 of file bbchvDaemon.cc.

```
14
15
16 CndDbSender* sender = new bbchvSender(ldir);
17
18 sender->initQuery();
19 for(;;) { //ever...
20     if(sender->hasBackLog())sender->cleanBackLog();
21     if(sender->queryData())sender->updateDb();
22     sleep(sender->sleepTime());
23 }
24
25 };
```

4.2 bbchvSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "bbchvSender.hh"
#include "StDbTable.h"
#include "bbchvSender_i.cc"
```

Defines

- #define __CLASS__ "bbchvSender"
- #define __METHOD__ "initTable()"
- #define __METHOD__ "initDataBase()"
- #define __METHOD__ "queryData()"
- #define __METHOD__ "readData(fileName)"
- #define __METHOD__ "updateDb(filename)"

4.2.1 Define Documentation

4.2.1.1 #define __CLASS__ "bbchvSender"

Definition at line 17 of file bbchvSender.cc.

4.2.1.2 #define __METHOD__ "updateDb(filename)"

4.2.1.3 #define __METHOD__ "readData(fileName)"

4.2.1.4 #define __METHOD__ "queryData()"

4.2.1.5 #define __METHOD__ "initDataBase()"

4.2.1.6 #define __METHOD__ "initTable()"

4.3 bbchvSender.hh File Reference

```
#include "CndDbSender.hh"
#include "bbchv.h"
```

Classes

- class [bbchvSender](#)

Defines

- #define [NUM_DB_ROWS](#) 64

4.3.1 Define Documentation

4.3.1.1 #define NUM_DB_ROWS 64

Definition at line 16 of file bbchvSender.hh.

4.4 bbchvSender_i.cc File Reference

Defines

- #define METHOD_ "loadUserControls(name,value)"
- #define METHOD_ "initQuery()"
- #define METHOD_ "readData(ifstream)"

4.4.1 Define Documentation

4.4.1.1 #define METHOD_ "readData(ifstream)"

4.4.1.2 #define METHOD_ "initQuery()"

4.4.1.3 #define METHOD_ "loadUserControls(name,value)"

4.5 fpdhvDaemon.cc File Reference

```
#include "fpdhvSender.hh"
#include <unistd.h>
```

Functions

- void **runSender** (const char **ldir*)

4.5.1 Function Documentation

4.5.1.1 void runSender (const char * *ldir*)

Definition at line 14 of file fpdhvDaemon.cc.

```
14
15
16     CndDbSender* sender = new fpdhvSender(ldir);
17
18     sender->initQuery();
19     for(;;) { /*ever...
20         if(sender->hasBackLog())sender->cleanBackLog();
21         if(sender->queryData())sender->updateDb();
22         sleep(sender->sleepTime());
23     }
24
25 };
```

4.6 fpdhvSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "fpdhvSender.hh"
#include "StDbTable.h"
#include "fpdhvSender_i.cc"
```

Defines

- #define __CLASS__ "fpdhvSender"
- #define __METHOD__ "initTable()"
- #define __METHOD__ "initDataBase()"
- #define __METHOD__ "queryData()"
- #define __METHOD__ "readData(fileName)"
- #define __METHOD__ "updateDb(filename)"

4.6.1 Define Documentation

4.6.1.1 #define __CLASS__ "fpdhvSender"

Definition at line 17 of file fpdhvSender.cc.

4.6.1.2 #define __METHOD__ "updateDb(filename)"

4.6.1.3 #define __METHOD__ "readData(fileName)"

4.6.1.4 #define __METHOD__ "queryData()"

4.6.1.5 #define __METHOD__ "initDataBase()"

4.6.1.6 #define __METHOD__ "initTable()"

4.7 fpdhvSender.hh File Reference

```
#include "CndDbSender.hh"
#include "fpdhv.h"
```

Classes

- class [fpdhvSender](#)

Defines

- #define [NUM_DB_ROWS](#) 304

4.7.1 Define Documentation

4.7.1.1 #define NUM_DB_ROWS 304

Definition at line 16 of file fpdhvSender.hh.

4.8 fpdhvSender_i.cc File Reference

Defines

- #define METHOD_ "loadUserControls(name,value)"
- #define METHOD_ "initQuery()"
- #define METHOD_ "readData(ifstream)"

4.8.1 Define Documentation

4.8.1.1 #define METHOD_ "readData(ifstream)"

4.8.1.2 #define METHOD_ "initQuery()"

4.8.1.3 #define METHOD_ "loadUserControls(name,value)"

Index

~bbchvSender
 bbchvSender, 6
~fpdhvSender
 fpdhvSender, 17
__CLASS__
 bbchvSender.cc, 28
 fpdhvSender.cc, 32
__METHOD__
 bbchvSender.cc, 28
 bbchvSender_i.cc, 30
 fpdhvSender.cc, 32
 fpdhvSender_i.cc, 34

bbchvDaemon.cc, 27
bbchvDaemon.cc
 runSender, 27
bbchvSender, 5
 bbchvSender, 6
bbchvSender
 ~bbchvSender, 6
 bbchvSender, 6
 cdriftLimit, 13
 elementList, 13
 hasChanged, 6
 initDataBase, 7
 initQuery, 7
 initTable, 8
 initTags, 8
 loadUserControls, 8
 mline, 14
 mreadStatus, 14
 nextLine, 9
 previousVals, 14
 ptr1, 14
 ptr2, 14
 queryData, 9
 readAny, 9
 readData, 10
 readError, 10
 readVal, 11, 12
 tempVals, 14
 tmpline, 14
 updateDb, 12
 updateElements, 14
 updateVals, 14

 vdriftLimit, 14
bbchvSender.cc, 28
bbchvSender.cc
 __CLASS__, 28
 __METHOD__, 28
bbchvSender.hh, 29
bbchvSender.hh
 NUM_DB_ROWS, 29
bbchvSender_i.cc, 30
bbchvSender_i.cc
 __METHOD__, 30

cdriftLimit
 bbchvSender, 13
 fpdhvSender, 24

elementList
 bbchvSender, 13
 fpdhvSender, 24

fpdhvDaemon.cc, 31
fpdhvDaemon.cc
 runSender, 31
fpdhvSender, 16
 fpdhvSender, 17
fpdhvSender
 ~fpdhvSender, 17
 cdriftLimit, 24
 elementList, 24
 fpdhvSender, 17
 hasChanged, 17
 initDataBase, 17
 initQuery, 18
 initTable, 18
 initTags, 19
 loadUserControls, 19
 mline, 25
 mreadStatus, 25
 nextLine, 20
 previousVals, 25
 ptr1, 25
 ptr2, 25
 queryData, 20
 readAny, 20
 readData, 20, 21

readError, 21
 readVal, 22, 23
 tempVals, 25
 tmpline, 25
 updateDb, 23
 updateElements, 25
 updateVals, 25
 vdriftLimit, 25
 fpdhvSender.cc, 32
 fpdhvSender.cc
 __CLASS__, 32
 __METHOD__, 32
 fpdhvSender.hh, 33
 fpdhvSender.hh
 NUM_DB_ROWS, 33
 fpdhvSender_i.cc, 34
 fpdhvSender_i.cc
 __METHOD__, 34
 hasChanged
 bbchvSender, 6
 fpdhvSender, 17
 initDataBase
 bbchvSender, 7
 fpdhvSender, 17
 initQuery
 bbchvSender, 7
 fpdhvSender, 18
 initTable
 bbchvSender, 8
 fpdhvSender, 18
 initTags
 bbchvSender, 8
 fpdhvSender, 19
 loadUserControls
 bbchvSender, 8
 fpdhvSender, 19
 mline
 bbchvSender, 14
 fpdhvSender, 25
 mreadStatus
 bbchvSender, 14
 fpdhvSender, 25
 nextLine
 bbchvSender, 9
 fpdhvSender, 20
 NUM_DB_ROWS
 bbchvSender.hh, 29
 fpdhvSender.hh, 33
 previousVals

bbchvSender, 14
 fpdhvSender, 25
 ptr1
 bbchvSender, 14
 fpdhvSender, 25
 ptr2
 bbchvSender, 14
 fpdhvSender, 25
 queryData
 bbchvSender, 9
 fpdhvSender, 20
 readAny
 bbchvSender, 9
 fpdhvSender, 20
 readData
 bbchvSender, 10
 fpdhvSender, 20, 21
 readError
 bbchvSender, 10
 fpdhvSender, 21
 readVal
 bbchvSender, 11, 12
 fpdhvSender, 22, 23
 runSender
 bbchvDaemon.cc, 27
 fpdhvDaemon.cc, 31
 tempVals
 bbchvSender, 14
 fpdhvSender, 25
 tmpline
 bbchvSender, 14
 fpdhvSender, 25
 updateDb
 bbchvSender, 12
 fpdhvSender, 23
 updateElements
 bbchvSender, 14
 fpdhvSender, 25
 updateVals
 bbchvSender, 14
 fpdhvSender, 25
 vdriftLimit
 bbchvSender, 14
 fpdhvSender, 25